SGML: SGML and HyTime

Quick Notes on
HyTime
ISO 10744

Subject: Re: SGML contra HyTime

Date: 9 Jan 1997 16:52:05 GMT

From: "W. Eliot Kimber" <drmacro@drmacro.com>

Newsgroup: comp.text.sgml

andreas knapp <knapp@iassnb.irp.uni-stuttgart.de> wrote in article
<5b29et$3cam@info4.rus.uni-stuttgart.de>...

```
> Hi all,
> =======
>
> is there anyone out there who could tell me the real difference between
> SGML and HyTime, yes, I know somehow HyTime is an SGML application, but
> what exactly does it do for me if I want to create Hypermedia documents
> automatically, would it be better for me to use SGML of HyTime?
```

[Eliot]

HyTime is an application of SGML: to oversimplify, HyTime defines a set of element types that you can use with your existing SGML document types to provide hyperlinking and other facilities in a standardized way.

In other words, HyTime effectively extends SGML by defining a standard set of facilities for doing hypertext and multimedia presentations.

HyTime does not compete with or change SGML in any way--it simply builds on the basic facilities SGML provides. In particular, HyTime provides the following hypertext-related facilities:

1. A general model of hyperlinks and syntax for hyperlink representation using SGML elements. This is much more than simple ID/IDREF.
2. Facilities for addressing things other than elements with IDs in the local document: cross-document ID refs, references to elements without IDs, references to data characters, references to multiple objects at once.

These are facilities that all hypertext applications need. HyTime provides a general and robust mechanism for adding them to SGML documents and SGML processing applications. HyTime can also be integrated with other mechanisms, such as TEI extended pointers or HTTP URLs.

There is much more to HyTime than this, but the above are probably of immediate interest and utility to most SGML document type designers.

Technicaly, HyTime is an "enabling architecture". In particular, rather than defining element types that you must use directly in your DTD (as you would if you used part of a standard DTD, say the 12083 Maths elements), it provides a set of 'meta-element types' that serve as templates or supertypes from which you derive your own element types with their own element type names and specialized attributes. As long as your element types conform to the minimum requirements of the HyTime-defined meta-element types ("architectural forms"), a HyTime-aware processor will be able to provide HyTime-related services, such as as finding the anchors of hyperlinks.

Say you want to provide a general hyperlinking element in your document type. To make it possible for a HyTime processor to process this hyperlink it has to be told that your linking element is "derived from" a HyTime-defined hyperlink element form. You make this association by using an attribute to name the HyTime element form your hyperlink element is derived from. For the HyTime standard, this attribue is normally called "HyTime". In your declarations you do something like this:

```
<!-- My personal hyperlinking element, derived from the HyTime
     "contextual link" (clink) form: -->
<!ELEMENT CrossRef   - - (#PCDATA) -- Cross reference -->
<!ATTLIST CrossRef
          linkend    IDREF #REQUIRED
          -- Names ID of element cross reference is to --
          HyTime     NAME  #FIXED "clink"
          -- Tell HyTime processor this is a HyTime clink --
>
```

The HyTime declaration for the contextual link element form is (more or less):

```
<!element clink   -- Contextual link --
                  - O (%ArcCFC;)*
                  -- Can contain any elements or data -->
<!attlist clink
          linkend   -- Address of other end of link --
                    IDREFS #REQUIRED
>
```

The rules for deriving element types from element forms are fairly simple: For each required attribute of the element form (clink, in this case), the dervied element (CrossRef) must have the same attribute with the same name and its value must be consistent with the architectural attributes value prescription. The content of the client must be consistent with the content model of the meta element type (in this case, clink can contain anything or nothing, so it's impossible to be inconsistent with it).

Thus, we can see that the CrossRef element meets these requirements because the clink form requires an attribute called "linkend" with a value prescription fo IDREFS and the CrossRef element has an attribute called "linkend" with a value prescription of IDREF. The attribute name is the same and the value prescription "IDREF" is consistent with IDREFS (every IDREF is a valid value for an IDREFS attribute).

The HyTime attribute of the CrossRef element indicates that this element type is derived from the HyTime clink element form--this is enough information for a HyTime processor, such as SoftQuad's Panorama or TechnoTeacher's HyBrowse, to know how to process the CrossRef element as a HyTime hyperlink.

If you want to use different attribute names in your element types, you can map the HyTime-defined names to your own names using the HyNames attribute. For example, your authors might find an attribute like "section" more intuitive than "linkend" for the CrossRef element. You can do this:

```
<!-- My personal hyperlinking element, derived from the HyTime
     "contextual link" (clink) form: -->
<!ELEMENT CrossRef   - - (#PCDATA) -- Cross reference -->
<!ATTLIST CrossRef
          section    IDREF #REQUIRED
          -- Names element cross reference is to, e.g., "Section" --
          HyTime     NAME  #FIXED "clink"
          -- Tell HyTime processor this is a HyTime clink --
          HyNames    CDATA #FIXED "linkend section"
          -- Map HyTime attribute name "linkend" to our preferred
             name "section" --
>
```

A HyTime processor will now know that the attribute "section" is really the HyTime attribute "linkend" and will know what to do (otherwise the HyTime processor would have no idea what the section attribute was for and would ignore it).

You can learn much more about HyTime from the TechnoTeacher Web site, "http://www.techno.com", and from my personal Web site, "http://www.drmacro.com/hyperlink". You can get a free version of Panorama from SoftQuad at "http://www.sq.com". You can also get my Perl HyTime library Hy-Lib from the ISOGEN Web site at "http://www.isogen.com/demos/hy-lib.html".

--
<Address HyTime=bibloc homepage="http://www.drmacro.com">
W. Eliot Kimber, eliot@isogen.com
Senior SGML Consulting Engineer, Highland Consulting
2200 North Lamar Street, Suite 230, Dallas, Texas 75202
+1-214-953-0004 +1-214-953-3152 (fax)
http://www.isogen.com (work)</Address>
"Rats in the morning, rats in the afternoon...if they don't go away,
I'll be reducated soon..." --Austin Lounge Lizards, "1984 Blues"
(http://www.webcom.com/~yeolde/all/lllhome.html)